Relational grounded language learning

Hendrik Blockeel[1]

joint work with Leonor Becerra-Bonache[2],
Maria Galvan[2], and François Jacquenet[2]

[1] KU Leuven [2] Université Jean Monnet, Saint-Etienne

Language learning

- How do children learn language?
- Can we make a computer learn language in a similar way?

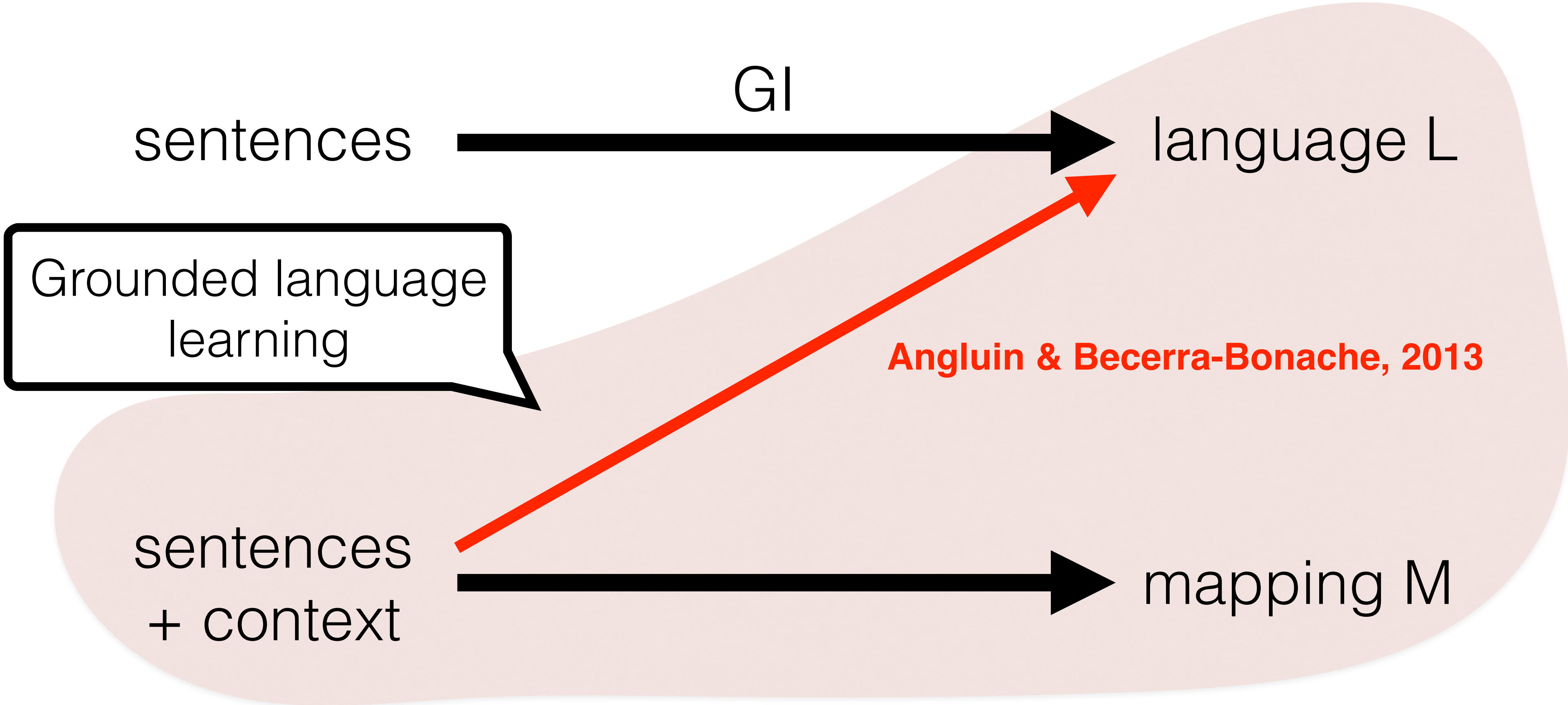
Language learning

- Grammatical inference: given examples of strings from a language, learn a grammar defining that language
- Much work on learning formal languages (regular, context-free, …) from examples
- But children do not learn in this manner!
  - they get as examples, language utterances *in a particular context*
  - by matching utterances to the context, they learn both *grammar* and *semantics* of languages

Different types of language learning

- <u>Grammatical inference</u>: learn a definition of the language from examples of sentences
  - language = set of strings
  - definition: via automaton, regular expression, context-free grammar, …
- <u>Grounded language learning</u>: from utterances of sentences in a physical context, learn a mapping between *sentence elements* and *things observed*
  - often focused on learning to map words to observable properties

Grounded language learning, broadly



Interested not only in how context helps with learning M, but also in how it helps with learning L

Different types of inputs

sentences

"the book is on the table"

much research

sentences
+ meaning

"the book is on the table"

is(book(b),on(table(t)))

e.g., learning semantic parsers

sentences
+ context

"the book is on the table"

Mooney et al., A-BB,
+ **this** research

book(b1). chair(c1). chair(c2).
table(t1). painting(p1). wall(w1).
on(b1,t1). on(p1,w1). …

Learning meanings:
Supervised vs. *weakly supervised*

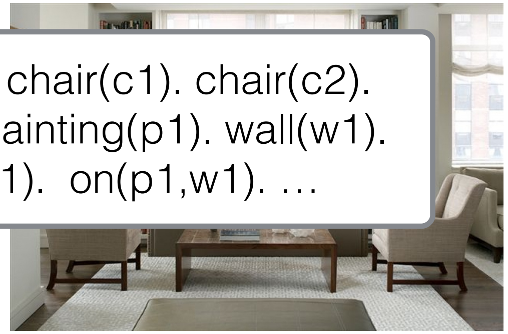
“this is the meaning”	“the meaning is somewhere in here”

“the book is on the table”

is(book(b),on(table(t)))

learning semantic parsers

“the book is on the table”



book(b1). chair(c1). chair(c2).
table(t1). painting(p1). wall(w1).
on(b1,t1). on(p1,w1). …

“that’s a red mug”

most grounded learning

λx.color(x,red)∧type(x,mug)

“I see a red mug”

some grounded learning

Grounded language learning

- For us, “grounded” language learning means:
  - the example sentences are grounded: linked to a particular context
  - the learned language model defines a set of correct sentences + the “meaning” of these sentences
  - No consensus on what “meaning” means! See Frege, Wittgenstein, Kripke, Harnad, …
  - Pragmatic solution: “any context in which it can be used” (Mooney, this work)

Angluin & Becerra-Bonache, 2013

- Study learning from (sentence, *context*) examples
- Present a model that understands & generates language utterances
- Use this model to study the effect of corrections on language learning
- *Model is relatively complex and ad-hoc: relies on first order logic, weighted graphs, decision trees, transducers, …*

This approach

- We propose a *principled approach* for representing *context* and *meaning* using first-order logic
- We propose a simple, incremental learning algorithm for learning a mapping between n-grams and meanings
  - no assumptions whatsoever about language structure, except that utterances are sequential
- We study its behavior in a simple "blocks world" (IDA) and later in a more challenging world (ECAI)

Becerra-Bonache, Blockeel, Galvan & Jacquenet, "A First-Order-Logic Based Model for Grounded Language Learning", in Proc. of IDA 2015, LNCS 9385: 49-60

Context

- A **context description** (briefly **context**) is a set of ground facts (often called a “model” or “interpretation” in inductive logic programming)

- An **n-gram** is a sequence of n words

- The **meaning** of an n-gram is whatever is in common among all the contexts in which that n-gram can be used

book(b1). chair(c1). chair(c2).
table(t1). painting(p1). wall(w1).
on(b1,t1). on(p1,w1). …

Assumptions

- Inputs consist of examples (context, phrase)
- Assumptions:
  - <u>Contexts are complete w.r.t. phrases</u> (phrase cannot refer to something not in the context)
  - <u>Phrases may be incomplete w.r.t. context</u> (not each context element needs to be mentioned)

“book” $\in$ phrase $\Rightarrow$ book $\in$ picture

chair $\in$ picture $\not\Rightarrow$ “chair” $\in$ phrase

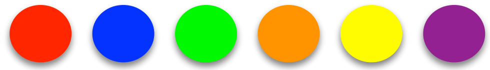
Context Vocabulary

- **Principle:** use predicates and constants of which it is plausible that they represent concepts that a child recognizes
  - e.g. object(o1) : there is some object (referred to as o1 here)
  - e.g. color(o1, red): the <u>color</u> of o1 is <u>red</u>
    - assumes the child recognizes <u>color</u> as being one aspect of visual appearance, <u>shape</u> as another aspect, etc.
    - assumes the child recognizes <u>red</u>, <u>blue</u>, … as different colors
    - compare to, e.g., “red(o1)”, “square(o1)”, …

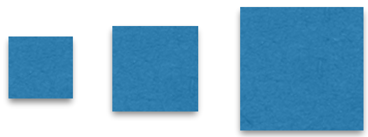
A toy world

- 6 shapes: sq, tr, di, st, he, el

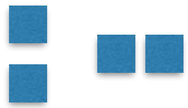
- 6 colors: re, bl, gr, or, ye, pu



- 3 sizes: sm, me, bg



- 4 relative positions: ab, be, lo, ro

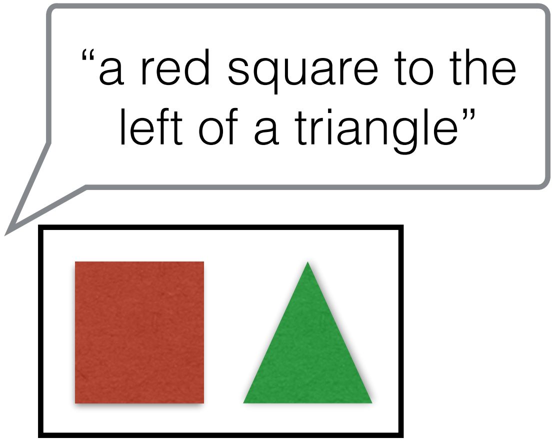


- 1 or 2 objects per context

The toy language

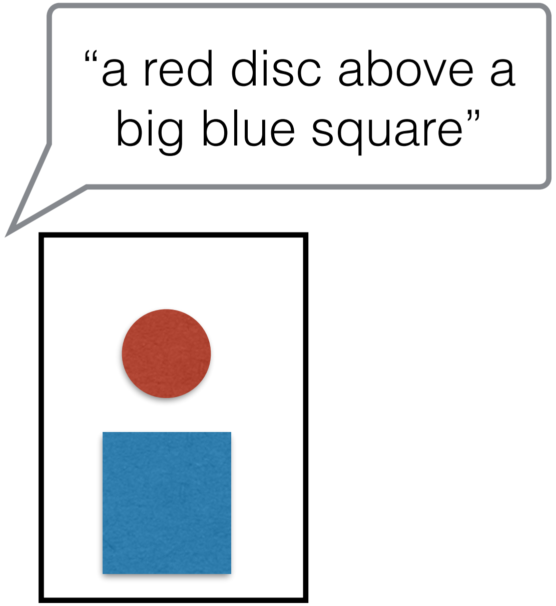
- words refer to <u>shapes</u>, <u>colors</u>, <u>sizes</u>, and <u>relative positions</u>
- phrases of the form
  - a big purple rectangle to the left of a small green disc
  - the purple triangle above the disc
  - …
- in 3 languages: English, Dutch, Spanish

Examples



{obj(1),clr(1,re),shp(1,sq),sz(1,bg),
obj(2),clr(2,gr),shp(2,tr),sz(2,bg), rp(1,lo,2)}

[a, red, square, to, the, left, of, a, triangle]



{obj(3),clr(3,re),shp(3,di),sz(3,me),
obj(4),clr(4,bl),shp(4,sq),sz(4,bg), rp(3,ab,4)}

[a, red, disc, above, a, big, blue, square]

Step 1: computing the meaning

- **Principle:** “The meaning of an n-gram is everything that is in common in all the contexts where it can be used”

- **Implementation:** via Plotkin’s (1970) “lgg” operator

  - intuitively: we find the most specific pattern (= existentially quantified conjunction) common to all contexts

  - technical definition relies on concepts of clause and theta-subsumption

Clauses

- A **clause** is a set of literals (usually interpreted as a universally quantified disjunction of literals)
  - set form: {father(X,Y), ¬parent(X,Y), ¬male(X)}
  - interpretation:
    - $\forall x,y$: father(x,y) $\vee$ ¬parent(x,y) $\vee$ ¬male(x)
    - $\forall x,y$: father(x,y) $\leftarrow$ parent(x,y) $\wedge$ male(x)

“head” “body”, = conjunction

Theta-subsumption

- A clause c **subsumes** a clause d iff there exists a variable substitution θ such that $c\theta \subseteq d$
- Examples:
  - {father(X,Y)} subsumes {father(john, mary)} (θ={X→john,Y→mary})
  - {father(X,Y), ¬male(X)} subsumes {father(john,Y), ¬male(john)} (θ={X→john})
  - father(X,Y) ← male(X) subsumes father(X,Y) ← parent(X,Y), male(X) (θ={})
- subsumes = “is more general than” ≃ entails

Theta-subsumption

- The least general generalization under theta-subsumption ("lgg") of two clauses = most specific clause that subsumes both
- Simple (though non-trivial) algorithm for computing it : Plotkin, 1970

used("red") ← obj(o1), clr(o1,re), shp(o1,sq), obj(o2), clr(o2,bl), shp(o2,tr)
used("red") ← obj(o3), clr(o3,or), shp(o3,sq), obj(o4), clr(o4,re), shp(o4,tr)

used("red") ← obj(X), clr(X,re), shp(X,_), obj(Y), clr(Y,_), shp(Y,tr),
obj(Z), clr(Z,_), shp(Z,sq)

Theta-subsumption

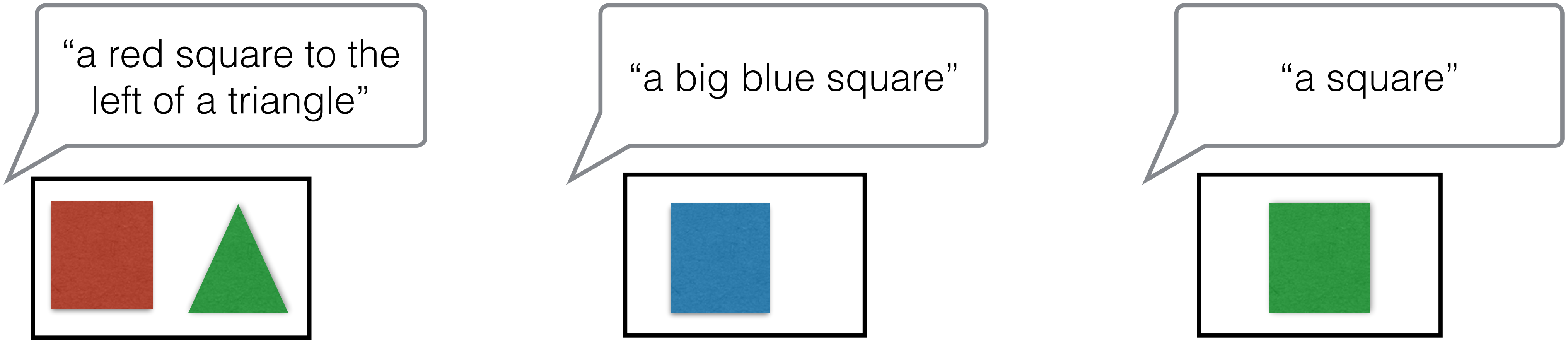
- When no variables in head of clause, can move quantifier to body (changing ∀ to ∃)
- Hence our interpretation of lgg as the *maximal common pattern*, where pattern = existentially quantified conjunction

$$\forall x,y,z: \text{used}(\text{"red"}) \leftarrow \text{obj}(x), \text{clr}(x,\text{re}), \text{sh}(x,\_), \text{obj}(y), \text{clr}(y,\_), \text{sh}(y,\text{tr}), \text{obj}(z), \text{clr}(z,\_), \text{sh}(z,\text{sq})$$

$$\Updownarrow$$

$$\text{used}(\text{"red"}) \leftarrow \exists x,y,z: \text{obj}(x), \text{clr}(x,\text{re}), \text{sh}(x,\_), \text{obj}(y), \text{clr}(y,\_), \text{sh}(y,\text{tr}), \text{obj}(z), \text{clr}(z,\_), \text{sh}(z,\text{sq})$$

Lgg of these contexts?

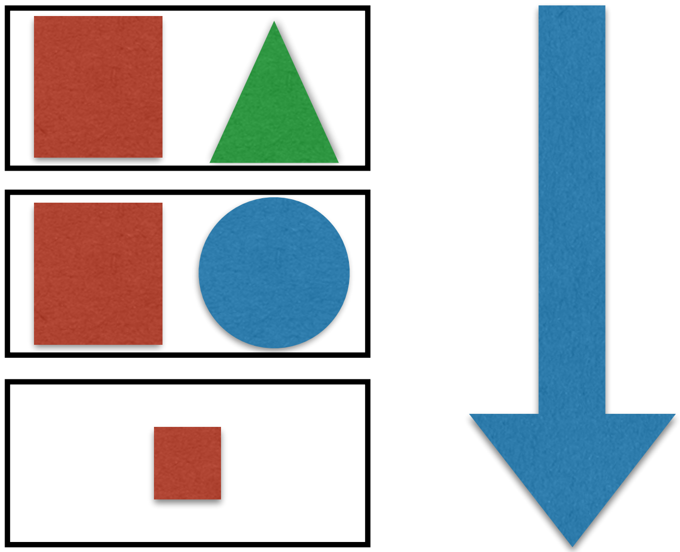


n-gram	lgg of context in which it appears
“square”	{obj(X), shp(X,sq), clr(X,_), sz(X,bg)}
“big”	{obj(3), shp(3,sq), clr(3,bl), sz(3,bg)}
“a”	{obj(X), shp(X,_), clr(X,_), sz(X,_)}

Computing the meaning

- Note: the more contexts, the less they have in common
- So the meaning of an n-gram starts out as something very specific, and gradually becomes more general

E.g., evolution of current meaning of n-gram “red square”:



{obj(1),shp(1,sq),clr(1,re),sz(1,bg),
obj(2),shp(2,tr),clr(2,gr),sz(2,bg),rp(1,lo,2)}

{obj(X),shp(X,sq),clr(X,re),sz(X,bg),
obj(Y),shp(Y,_),clr(Y,_),sz(Y,bg),rp(X,lo,Y)}

{obj(X),shp(X,sq),clr(X,re),sz(X,_)}

Learning the meaning of n-grams

```
whenever a new example (C,S) is presented do:
    for all n-grams G in S, n=1, 2, …:
        updateMeaning(G,C)

updateMeaning(G,C):
    if meaning(G) undefined then meaning(G)=C
    else meaning(G) <- lgg(C, meaning(G))
```

The Prolog version

```
whenever a new example (C,S) is presented do:
    for all n-grams G in S, n=1, 2, ...:
        updateMeaning(G,C)

updateMeaning(G,C):
    if meaning(G,M) then
        retract meaning(G, M)
        assert meaning(G, lgg(C, M))
    else
        assert meaning(G,C)
```

Step 2: linking words to particular constants

- **Principle:** *it is useful to have words that refer to a particular physical concept*
    - E.g., useful to be able to refer to the color "red" (and other constants)
    - In fact, this is the first motivation for using language: being able to refer to things
- **Implementation:** build a mapping that links words to specific constants in context description
    - If the meaning of a word w contains exactly one constant c, and we assume there is a word for that constant, then it is likely that w is that word
    - In such cases, the learner concludes w may refer to c: **mrf(w,c)**

Linking words to particular constants

- Don’t jump to conclusions too fast… how do we know that our current belief about the meaning of w has converged?

- Rule: if the meaning has remained unchanged for the last *s* updates (s = “stability parameter”), assume it’s final

- At that point, if there is exactly one constant c in meaning, assert mrf(w,c)

- Should the meaning change anyway, afterwards, then retract mrf(w,c) again

Algorithm, v2

```
whenever a new example (C,S) is presented do:
    for all n-grams G in S, n=1, 2, …:
        updateMeaning(G,C)

updateMeaning(G,C):
    if meaning(G,M) then
        if lgg(C,M)=M then
            stab(G)++;
            if G is 1-gram & stab(G)=s & M contains one constant c
            then assert mrf(G,c)
        else
            retract meaning(G, M);
            assert meaning(G, lgg(C, M));
            stab(G) = 0;
            if mrf(G,X) then retract mrf(G,X)
    else assert meaning(G,C); stab(G) = 0
```

Step 3: generalizing n-grams

- **Principle**: it is likely that a learner that recognizes the concepts color, shape, … at some points sees a pattern: “red square”, “yellow triangle”, … -> “*color shape*”
  - All that’s needed for this is the tendency to *categorize* & to *generalize* (from element to category)
  - Plenty of evidence that children (even animals) do this
- **Implementation**: store category(v,p,i) <=> v occurs as i’th argument of p
  - e.g., category(re,clr,2) : “red is a color"

Step 3: generalizing n-grams

- **Implementation:** We allow our learner to generalize n-grams, for n>1, as follows (for simplicity we illustrate on bigrams)

- Consider a fact meaning([w_1, w_2], M) for which mrf(w_1, c_1) and mrf(w_2, c_2) exist, with category(c_1, p_1, a_1) and category(c_2, p_2, a_2)

- Make a rule meaning([W_1, W_2], M') :- mrf(W_1,C_1), mrf(W_2, C_2), category(C_1, p_1, a_1), category(C_2, p_2, a_2) where M' is M with c_1, c_2 changed into C_1, C_2

- *Evaluate rule*; if good, assert it and retract all facts implied by it

[red,square] -> obj(X),clr(X,re),shp(X,sq),sz(X,_)	red -> re square -> sq	category(re,clr,2) category(sq,shp,2)
[blue,triangle] -> obj(X),clr(X,bl),shp(X,tr),sz(X,_)	blue -> bl triangle -> tr	category(bl,clr,2) category(tr,shp,2)

[C,S] -> obj(X),clr(X,C'),shp(X,S'),sz(X,_)	if	C -> C' S -> S'	category(C,clr,2) category(S,shp,2)

Evaluating the quality of a rule

- If rule predicts a different meaning than the currently stored meaning, is the rule wrong?
- Not necessarily: perhaps the meaning stored in that fact has not yet converged
- We distinguish **three cases**:
  - **E**: set of n-grams for which rule predicts the currently stored meaning (stored meanings **confirm** rule)
  - **S**: set of n-grams for which the rule predicts a generalization of the currently stored meaning (stored meanings **compatible** with rule); $E \subseteq S$
  - **I**: set of n-grams for which the rule's prediction is not a generalization (stored meanings for these n-grams **contradict** rule)

Evaluating the quality of a rule

- A rule is **valid** if I is empty (no contradictions)
- |E| is a measure for the **evidence** in favor of the rule
- |S| is a measure for the **usefulness** of the rule
- In our implementation, we add the rule if it is valid and has enough evidence: $|E| \geq e$ with e the "evidence parameter"
- Note: adding such a rule ``boosts convergence'' for facts in S \ E

Illustration

for brevity:

{obj(o_1),shp(o_1,sq),clr(o_1,_),sz(o_1,bg)} —> [o_1,bg,sq], etc.

[bl,sq] real meaning (lgg fixpoint)

[bg,bl,sq]

[o_1,bg,bl,sq] after seeing 1 context where bigram “blue square” appeared

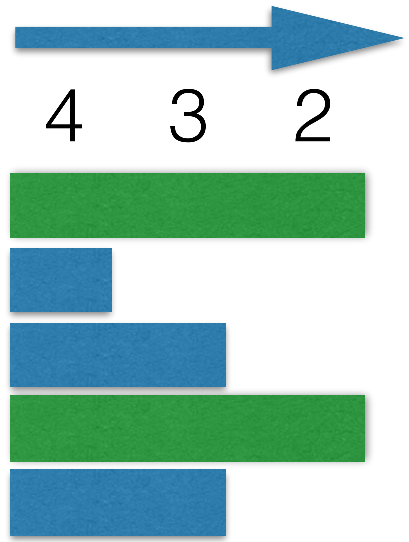
“blue square”

“current” meaning gradually becomes more general

Illustration

generalization level
(#constants left)

evidence
compatible



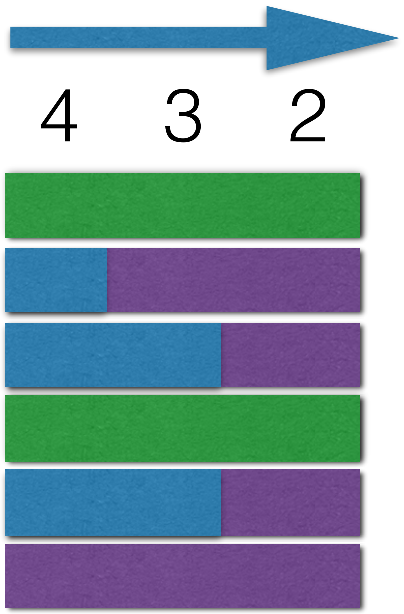
"blue square"	{obj(X),shp(X,**sq**),clr(X,**bl**),sz(X,_)}
"red square"	{obj(**6**),shp(**6**,**sq**),clr(**6**,**re**),sz(**6**,**sm**)}
"green triangle"	{obj(X),shp(X,**tr**),clr(X,**gr**),sz(X,**sm**)}
"purple disc"	{obj(X),shp(X,**di**),clr(X,**pu**),sz(X,_)}
"green square"	{obj(X),shp(X,**sq**),clr(X,**gr**),sz(X,**bg**)}
"purple triangle"	

rule:
if mrf(W_1,C_1) & mrf(W_2,C_2) & C_1 is a color & C_2 is a shape
then "W_1 W_2" means {obj(X),shp(X,C_1),clr(X,C_2),sz(X,_)}

Illustration

generalization level
(#constants left)

“boosted”



“blue square”	{obj(X),shp(X,**sq**),clr(X,**bl**),sz(X,_)}
“red square”	{obj(X),shp(X,**sq**),clr(X,**re**),sz(X,_)}
“green triangle”	{obj(X),shp(X,**tr**),clr(X,**gr**),sz(X,_)}
“purple disc”	{obj(X),shp(X,**di**),clr(X,**pu**),sz(X,_)}
“green square”	{obj(X),shp(X,**sq**),clr(X,**gr**),sz(X,_)}
“purple triangle”	{obj(X),shp(X,**tr**),clr(X,**pu**),sz(X,_)}

rule:
if mrf(W_1,C_1) & mrf(W_2,C_2) & C_1 is a color & C_2 is a shape
then “W_1 W_2” means {obj(X),shp(X,C_1),clr(X,C_2),sz(X,_)}

Algorithm, v3

whenever a new example (C,S) is presented **do**:
    for all n-grams G in S, n=1, 2, …:
        updateMeaning(G,C)
        generalize(G)

updateMeaning(G,C): as before, except: call cleanup after asserting mrf

generalize(G):
    call meaning(G, M)
    Ref = {mrf(w,c) | w in G)}
    Cat = {category(c,p,i) | mrf(_,c) in Ref}
    R = meaning(G, M) :- Ref, Cat
    for each w_i, c_i that occurs in Ref, introduce a new variable W_i, C_i
    change all occurrences of w_i, c_i in R into W_i, C_i
    if R is valid and evidence(R) ≥ e **then** assert(R); cleanup

cleanup: remove all facts covered by some rule R

Some experiments

- We have given the system 1000 examples of contexts & phrases for each of 3 languages (English, Dutch, Spanish)
  - randomly generated using a simple grammar
- Studied the “learning curve”
- Checked usefulness of learned model for understanding & generating phrases

Learning curve

79: the "*color shape*" generalized bigram is learned. The mrf map at this point contains 4 colors and 5 shapes, hence the rule predicts the meaning of 20 combinations. 5 predictions are equivalent to the stored meaning, 14 generalize it ("boosting convergence"), 1 is for a bigram not seen before.
85: mrf(to,bg) is retracted. Apparently, the system had earlier concluded that "to" means bg (big), because that was the only constant common in all its contexts and it remained present in the next 5 contexts. When finally a context for "to" without a big object is seen, bg disappears from the meaning; it is then clear that mrf(to,bg) was added prematurely, and it is retracted.
86: mrf(disc,di) is added. The meanings of "red disc" and "blue disc" are retracted, as they are now subsumed by the color-shape rule.
165: the "***size color***" generalized bigram is learned.
188: "$start the *color shape* $stop" is learned (this pattern forms a **full phrase**)
416: the "*size color shape*" generalized **trigram** is learned.
664: "*shape* to the *relpos* of" is learned. This is an overgeneralization: it correctly covers the words "left" and "right", but incorrectly also "above", "below" and "under", all of which are associated with relative positions.

Language peculiarities learned

- Learns that in English, you typically say “big blue square”, not “blue big square”
- Learns that in French, you say “grand triangle bleu” (size before noun, color after noun)
- Learns that in Spanish, you say “triangulo azul y grande” (using connective “y”)

Using the model…

- Given a context, generate relevant phrases

{object(90), shape(90,tr), color(90,re), size(90,me), object(91), shape(91,he), color(91,pu), size(91,bg), rel pos(90,lo,91)}

→ "the red triangle", "the triangle to the left of the hexagon", …

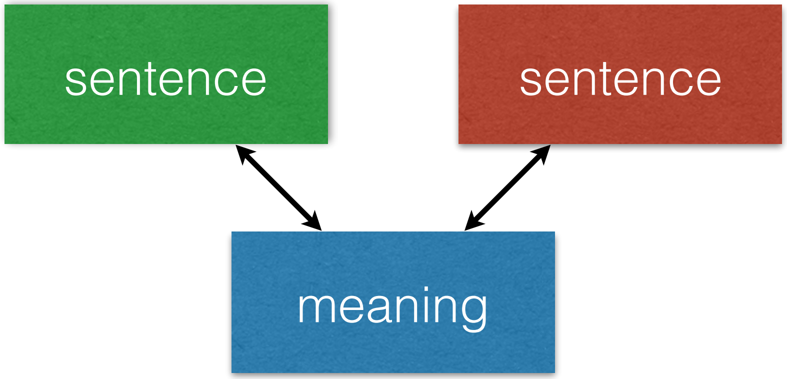
- Given a phrase and a context, identify the part of the context the phrase talks about

"the red triangle" {object(90), shape(90,tr), color(90,re), size(90,me), object(91), shape(91,he), color(91,pu), size(91,bg), rel pos(90,lo,91)}

→ {**object(90), shape(90,tr), color(90,re),** size(90,me), object(91), shape(91,he), color(91,pu), size(91,bg), rel pos(90,lo,91)}

Using the model…

- Translate a phrase to another language




```
?- meaning(ngram(4, [the, big, blue, triangle]), _, C1),
   meaning(L, spanish, C2), equiv(C1,C2).
```

```
L = ngram(5, [el,triangulo,azul,y,grande]);
L = ngram(5, [un,triangulo,azul,y,grande]);
…
```

No parallel corpora needed to “learn to translate”.
Contexts for English sentences ≠ contexts for Spanish sentences

Sometimes it gets it wrong…

- For Dutch sentences, among 6 shapes, only one (“vierkant”, = square) uses the determinate article “het”, all others use “de”
- => “het” is assumed to refer to a square
- => “het groene vierkant” gets generalized to “shape color shape”
- => the system generates “driehoek rode driehoek” when it sees a red triangle, etc.

A more realistic dataset

- IDA work used a specially prepared toy dataset
  - small world, few objects
  - limited vocabulary, simple sentences (generated automatically using a very simple grammar)
- What happens when we use a more challenging dataset, one not prepared specifically for this task?

Becerra-Bonache, Blockeel, Galvan & Jacquenet, "Relational grounded language learning", in Proc. of ECAI 2016
Becerra-Bonache, Blockeel, Galvan & Jacquenet, "Learning language models from images with ReGLL", in Proc. of ECMLPKDD 2016 (demo paper)

A more challenging dataset

- Zitnick et al., 2013: dataset of “clip-art” pictures with sentences commenting on what’s in the picture
  - many more objects, much more extensive vocabulary
  - “real” sentences, not generated by some simple grammar

[$start, mike, is, kicking, the, ball, $stop]

[object(o1), sky(o1, sun), color(o1, yellow), size(o1,big), …, object(o3), human(o3,boy), pose(o3,pose2), expression(o3,happy), object(o4), human(o4,girl), pose(o4,pose3), expression(o4,surprised), …, object(o6), clothing(o6,glasses), color(o6,violet), object(o7), toy(o7,ball), sport(o7,soccer), act(o3,wear,o6), …]

Problem 1: violation of main assumption

- IDA method assumed: sentence only mentions things present in the picture
- This dataset has exceptions. E.g.: “Mike is in the tent” - for a picture that shows a tent but doesn’t show Mike
- “lgg computation” does not allow for any exceptions
- **Quick fix**: compute lgg of a random subset of contexts that covers *most* (not necessarily all) contexts

Problem 2: learning the reference function

- IDA method: if the meaning of a word contains only one constant, assume that's what it refers to
- Does not work with these richer descriptions: sometimes too many constants remain in the meaning, sometimes all constants are gone, …
- Solution: <u>find the one constant that best "correlates" with word</u>
- Note: **asymmetric** measure of correlation needed!
  - both present / both absent increases correlation
  - *word without constant* decreases it much more than *constant without word*

	const	-const
word	high	~0
-word	?	high

Problem 2: solution

- **Solution**: use F_β-measure
- word = "prediction", constant = observation
- precision P = #(word&constant) / #word
- recall R = #(word&constant) / #constant
- F_1 = harmonic mean of P and R
- β <1 : precision more important than recall

$$F_\beta = (1+\beta^2)\frac{PR}{\beta^2 P+R}$$

	const	-const
word	A	B
-word	C	D

P=A/(A+B)

R=A/(A+C)

Problem 3: generation of sentences

- Often many sentences can be generated
- Try to generate the most interesting ones
- Criterion: frequency of n-grams occurring in the sentence (lower frequency = more specific)

Experimental results

- Learning from 10,000 (sentence,context) examples from Zitnick dataset
- Tested:
  - word -> constant mapping
  - sentence generation for previously unseen pictures

Mapping words-constants

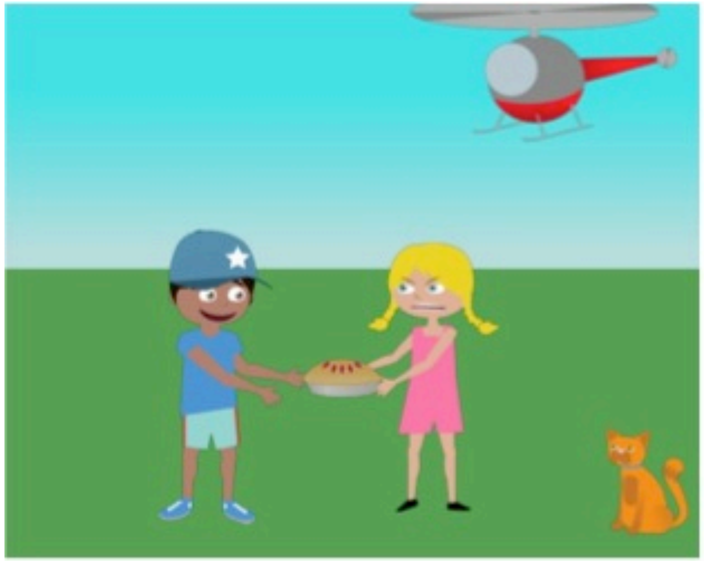
jenny	
mike	
to	
and	
hat	
owl	

ball	
tent	
cat	
sandbox	
table	
balloons	
fire	
ground	

angry	
snake	
grill	
helicopter	
raining	
slide	
duck	
bat	

bear	
glasses	
pizza	
balloon	
rocket	
crown	
lightning	
hotdogs	
shovel	

Sentence generation: some examples



1. the cat is sitting next to jenny
2. jenny is holding a pie
3. mike is wearing a blue cap
4. mike is wearing a blue hat
5. mike is wearing a hat

1. mike is wearing a pirate hat
2. jenny is in the sandbox
3. mike is kicking the soccer ball
4. jenny is angry at mike
5. mike is in the sandbox

No generalized n-grams

- In the “blocks world”, a meaning could be assigned to generalized n-grams ([a, X, square] …)
- Here, no such results were obtained
- Reason: too much variation in the “meaning” of similar words to find a rule

```
[mike, is, eating] ->
    [object(_A),color(_A,_C),object(_B),food(_B,_D),
     object(_E),human(_E,c_boy),pose(_E,_F),expression(_E,_G)]   % 34/36

[jenny, is, eating] ->
    [object(_A),object(_B),human(_A,_C),pose(_A,_D),expression(_A,_E),act(_A,_F,_B),
     object(_G),human(_G,c_girl),pose(_G,_H),expression(_G,_E),object(_I),food(_I,_J)]  % 12/13
```

Summarizing…

- To our knowledge, this is the first system to learn semantics and syntactical structure of language in a *weakly supervised* manner

- Still very preliminary, but leads to interesting insights and yields some promising results

- Model is very versatile: can be used for identification, description, translation, …

Future work…

- Learn to relate meaning of sentence to meanings of constituent parts
  - Learning on one hierarchical level will speed up learning on another level
  - Should speed up learning of generalized n-grams
  - Should naturally lead to a hierarchical grammatical structure
- Include words themselves into the context (e.g., grammatical gender)
- Include the discourse itself into the context (e.g., “the” vs. “a”)

Long-term perspectives…

- Results may shed some light on how language learning from positives only is possible

- Chomsky: "humans must possess a special-purpose "language learning device" that defines a universal grammar (of which only the parameters are filled in)"

- I believe that the ability to *categorize* and *generalize*, + the fact that we learn in a *context,* + the fact that language utterances are *sequential*, explains similarities & differences among languages.

- I hope that this research will ultimately demonstrate this.